5560 Business Intelligence and Data Visualization

Master of Science in Computing and Data Analytics

Department of Mathematics and Computing Science

High-Dimensional Analysis

Group 1:
Bhavik Kantilal Bhagat (A00494758)
Miguel Angel Palafox Gomez (A00494183)
Md Chistia Chowdhury (A00485184)

Date: December 15, 2025

Contents

1 Overview of High-Dimensional Analysis **3**
1.1 Challenges in Analysis . . . 3
1.2 Databases vs Spreadsheets . . . 4

2 Business Intelligence Strategy **6**
2.1 Customer Side Strategy . . . 6
2.2 Supplier Side Strategy . . . 6

3 Customer Side Analysis (Demand) **8**
3.1 Customer Data-cube . . . 8
3.2 Pivot Table Operations . . . 9
3.3 Visualization . . . 12
3.3.1 Bubble Chart . . . 12
3.3.2 Pie Chart . . . 14
3.3.3 Pareto Analysis . . . 14
3.3.4 Bar Chart . . . 16

4 Supplier Side Analysis (Supply) **18**
4.1 Supplier Data-cube . . . 18
4.2 Pivot Table Operations . . . 19
4.3 SQL Query . . . 19
4.4 Statistical Analysis . . . 22

5 Summary **26**
5.1 Customer Domain . . . 26
5.2 Supplier Domain . . . 26

6 Learning Reflection **28**
6.1 Multi-Dimensional Data Analysis . . . 28
6.2 Business Question . . . 28
6.3 The Synergy of Hybrid Analytical Techniques . . . 28
6.3.1 Visualization (Descriptive Analysis) . . . 28
6.3.2 SQL Queries (Diagnostic Analysis) . . . 29
6.3.3 Statistical Analysis (Prescriptive Analysis) . . . 29

7 Appendix **30**
7.1 Tables . . . 30
7.1.1 Customer Share in Revenue . . . 30
7.1.2 Revenue by Category . . . 30
7.1.3 Revenue by Product . . . 31
7.1.4 Supplier Pivot Table . . . 32

7.1.5 Supplier and Products 33
7.2 Scripts 35
7.2.1 SQL Script 35
7.2.2 Clustering Script 36

Bibliography **38**

1 Overview of High-Dimensional Analysis

In recent years, organizations have accumulated vast volumes of data, making large-scale data analysis a critical challenge. Big data is characterized by **volume** (the scale of data), **velocity** (the speed at which data is generated), and **veracity** (the accuracy and reliability of data). Data analytics involves **multidimensional data** used to describe facts related to **who, what, when, where, how, and why**. The dimensions represent various perspectives from which data can be analyzed.

1.1 Challenges in High-Dimensional Data Analysis

Despite having rich information, high-dimensional data presents significant challenges that can hinder effective data analysis. Figure 1 illustrates key challenges frequently encountered during high-dimensional analysis. [9]

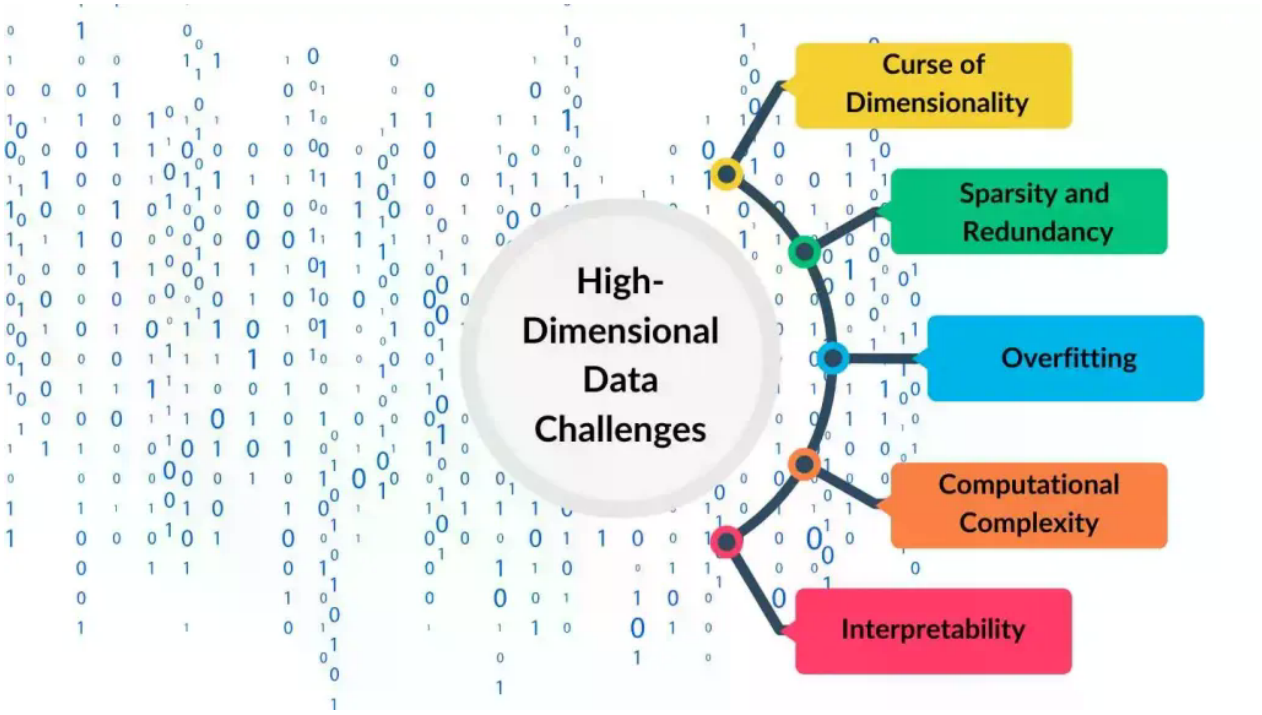


Figure 1: Some Challenges in High-Dimensional Analysis

- **Curse of Dimensionality**:
  This term describes the exponential increase in data volume and complexity as the number of dimensions increases. With many dimensions, it becomes very difficult to identify meaningful patterns and trends visually. Additionally, conventional distance metrics (e.g., Euclidean Distance) often become ineffective, which complicates statistical analysis for tasks like clustering or classification.

- **Sparsity and Redundancy**:
  In high-dimensional datasets, many features are often irrelevant, redundant, or empty. **Sparsity** occurs when only a few features are relevant for a specific observation; the rest of the empty or near-zero data can introduce noise. **Redundancy** occurs when the same type of information is captured by multiple features (multicollinearity), which introduces inefficiency and does not contribute much new value to the model.
- **Overfitting**:
  With an abundance of features relative to the number of data points, models can easily **overfit** the training data, capturing noise instead of generalized, meaningful patterns. That can lead to poor predictive performance due to reduced generality when applying the model to unknown data.
- **Computational Complexity**:
  The large number of features significantly increases the computational requirements to store, process, and analyze the data. This high complexity can substantially increase the time and cost associated with building models or conducting Exploratory Data Analysis (EDA).
- **Interpretability**:
  As the number of features grows, understanding the complex relationships and interactions in the data becomes much harder. This complexity makes **Explainable AI** (XAI) difficult, hindering the focus on transparency and interpretability of the model's decision-making process, which is often crucial in regulated sectors like healthcare or finance.

1.2 Why Database over Flat Spreadsheets?

While they provide a convenient way to view pre-processed data, they are insufficient for comprehensive high-dimensional analysis. Transitioning from flat files to a relational database structure proved essential for this project for the following reasons:

- **Data Integrity and Reduction of Redundancy:**
  Flat files often suffer from data redundancy, where the same fact is repeated multiple times, leading to inconsistencies and the potential for a "garbage can" effect in data storage. By utilizing a relational database, we leverage normalization through Primary Keys and Foreign Keys. This ensures that entities such as Customers, Products, and Suppliers are stored efficiently without unnecessary duplication, maintaining the integrity of the dataset.
- **Elimination of Data-Creator Bias:**
  A flat datasheet represents a pre-processed extraction of data, meaning the "Data-Creator" has already decided which columns and rows are relevant. This introduces bias and limits the scope of analysis. By accessing the underlying database directly, we were able to implement our own Business Intelligence (BI) strategy, retrieving

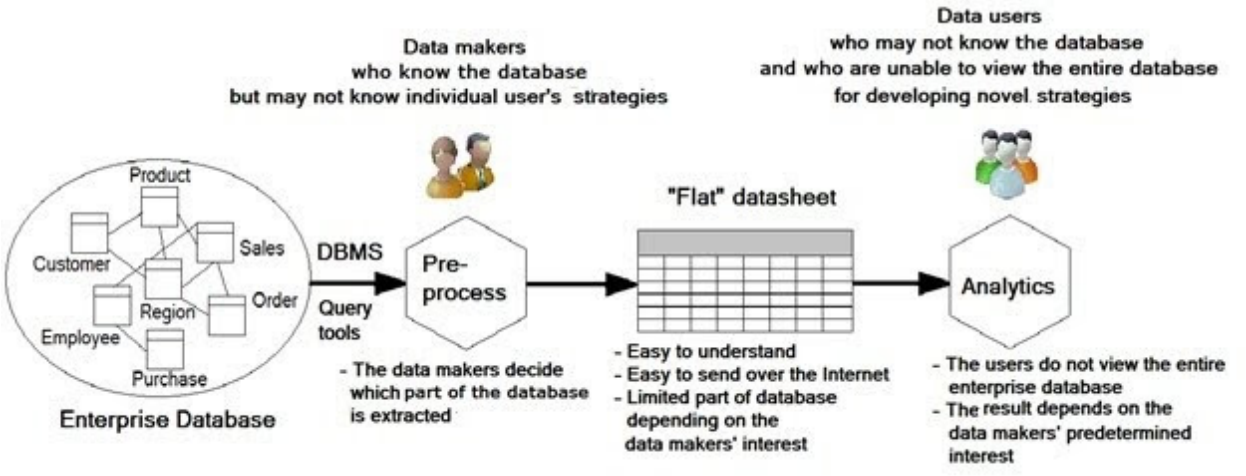


Figure 2: Why databases are preferred over flat Spreadsheets?

and connecting specific tables to reveal hidden patterns that a pre-constructed flat file might obscure.

- **Handling High-Dimensional Relationships:**
  Business strategies are rarely dependent on two-dimensional questions. A flat sheet is limited to rows and columns, but our analysis required simultaneous investigation of multiple dimensions (e.g., *Who* bought *What*, *When*, and from *Whom*). The database structure allowed us to construct Snowflake/Star Schema and Data Cubes, facilitating multi-level slicing, dicing, and drilling down into the data—capabilities that are severely restricted in a flat spreadsheet.
- **Managing Complex Relationships:**
  The sample data involves complex Many-to-Many (m:m) relationships, such as Students taking multiple Courses. In a relational database, these are resolved into One-to-Many (1:m) relationships using associative entities (e.g., Order Details). Understanding these linkages is necessary to accurately calculate measures like total revenue based on quantity and unit price across different hierarchical levels, a task that is prone to error in manual spreadsheet processing.

2 Business Intelligence Strategy

To ensure that the analysis yields actionable insights rather than generic observations, a targeted Business Intelligence (BI) strategy was developed. This strategy was systematically divided into two primary domains: the **Customer (Demand) Side** and the **Supplier (Supply) Side**. This structure guided the selection of high-dimensional analysis techniques, ensuring that every pivot table, query, visualization, and statistical analysis addressed a specific business risk or opportunity as per Figure 3.

2.1 Customer Side Strategy: Mitigating Revenue Dependence Risk

The primary objective on the demand side was to evaluate the stability of the company's revenue streams. Rather than simply listing sales figures, the strategy was focused on identifying revenue concentration risks and product demand cycles.

- **High-Dimensional Slicing of Demand Patterns:**
  To move beyond traditional analysis, three dimensions were utilized: *Who* (Customer), *What* (Product), and *When* (Time). Specifically, the Time dimension was used to track purchasing behaviors across Quarters and Months, allowing drill-down into specific transactions (e.g., identifying which products drove high demand in Q1 versus Q2).
- **Analysis of Revenue Concentration (Pareto Principle):**
  The aim was to determine if the company was relying disproportionately on a small subset of customers. By calculating the percentage contribution of each customer to total revenue, "Key Account" risks are identified, where the loss of a single client could threaten the financial stability of the business.
- **Product Portfolio Performance:**
  Visualizations were employed to simultaneously analyze three variables: Product, Order Quantity, and Time Period. This strategy was designed to highlight not just the best-selling products, but also the specific timing of those sales to optimize inventory planning.

2.2 Supplier Side Strategy: Cost Optimization and Arbitrage

The objective on the supply side was to reduce procurement costs by identifying price variances among suppliers for identical products. After finding suppliers who would charge above average, identifying alternative suppliers is critical for negotiating more favorable procurement terms.

- **Identification of Price Variance (SQL Query Strategy):**
  A core component of the strategy was the detection of "overcharging" suppliers. Specific SQL queries were designed to compare the *Unit Cost* offered by a particular sup-

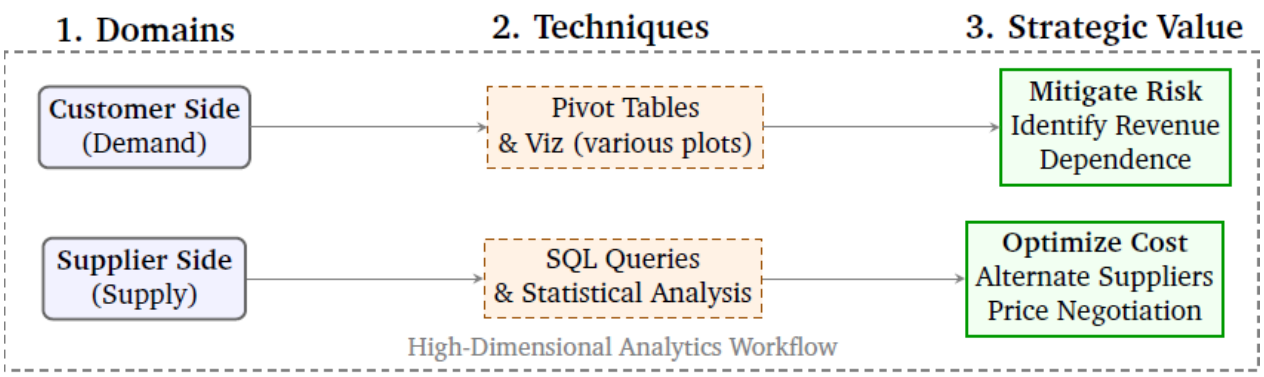


Figure 3: The Business Analytics Strategy map.Two primary business goals: 1. Reducing customer revenue dependence, and 2. Identifying overcharging suppliers.

plier against the *Average Market Price* for that product. The goal was to flag suppliers selling products significantly above the average, thus providing concrete targets for price negotiation.

- **Statistical Analysis of Cost Economies:**
  Hierarchical clustering analysis was also investigated to study the hierarchy among suppliers. The strategic goal was to determine if alternative suppliers could be found to procure products at a better price structure for the business.

3 Customer Side Analysis (Demand)

In alignment with the established BI strategy, the analysis was initiated on the demand side to investigate revenue dependence and product popularity. This process transitioned from structural definition (Data Cube, See Figure 4) to exploration (Pivot Tables) and finally to communication (Visualization).

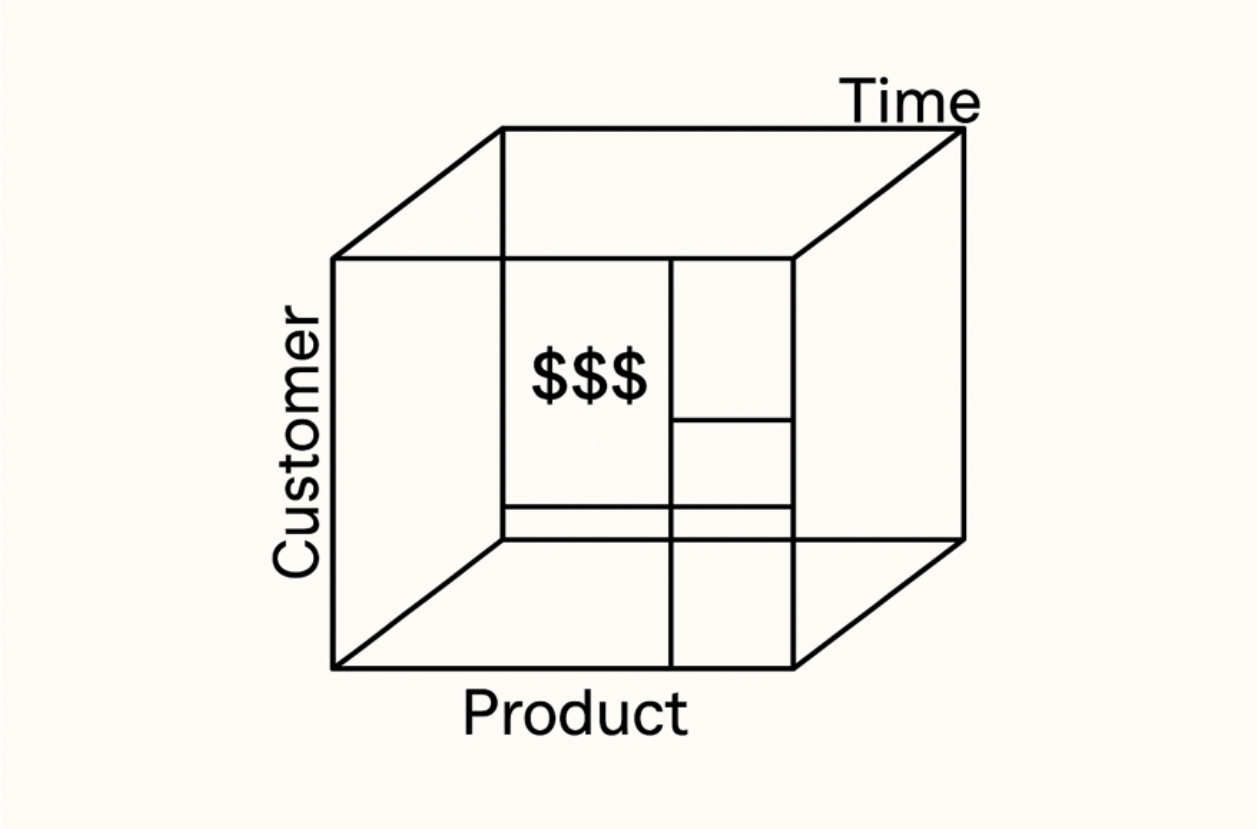


Figure 4: Customer Data-cube: Customer, Product, Order Date(Time)

3.1 Construction of the Demand-Side Data Cube

To enable high-dimensional analysis, a Snowflake Schema Figure 5 focused on the purchasing behavior of clients was constructed. As defined in the methodology, the central fact table and its surrounding dimensions were identified as follows:

- **Fact Table:**
  `Order Details` (containing measures: *Quantity* and *UnitPrice*).
- **Dimension Tables:**
  `Orders`(When), `Products`(What), & `Customers`(Who).

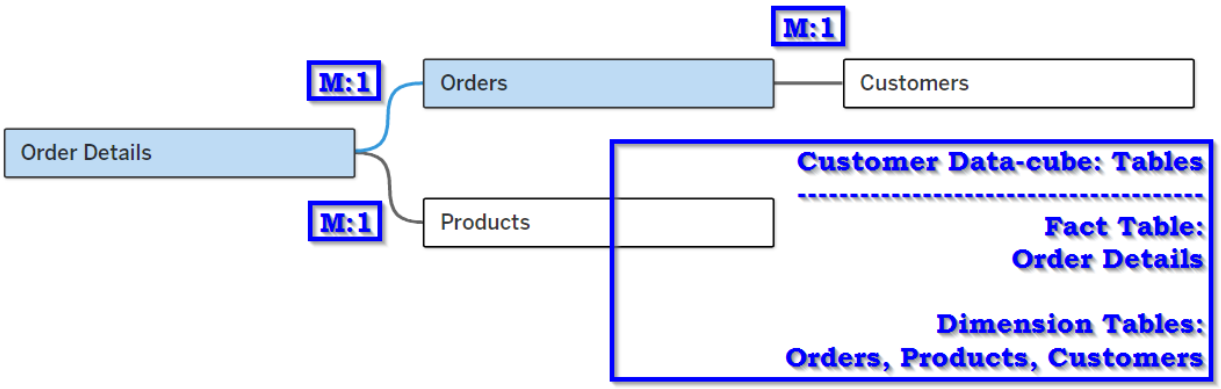


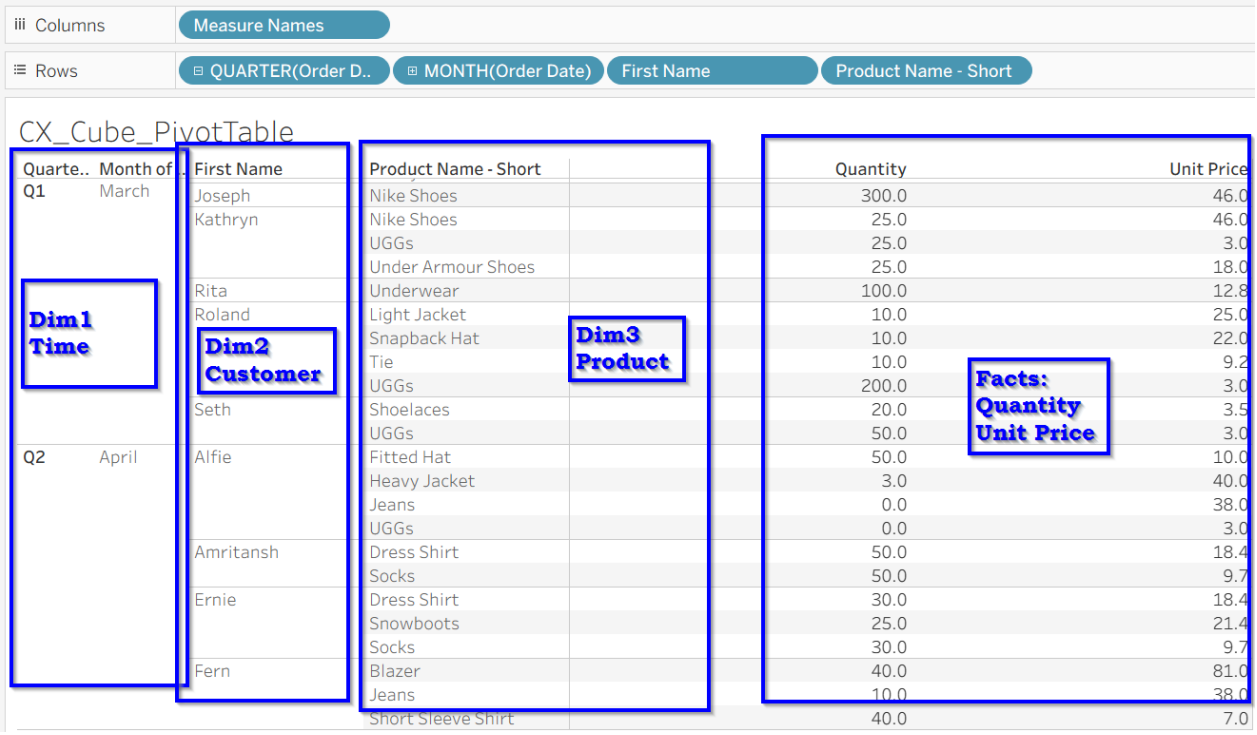
Figure 5: Customer Data-cube: Snowflake Schema Tables

These tables were linked via Primary and Foreign Keys (e.g., `CustomerID`, `ProductID`) to create a multi-dimensional structure capable of supporting the slicing and dicing operations required to answer complex questions about revenue concentration. The **Order Details** table and **Orders** table have a many-to-one (M:1) relationship, with `OrderID` acting as a foreign key for the **Order Details** table. The **Order Details** table and the **Products** table have a many-to-one (M:1) relationship, with `ProductID` acting as the foreign key. Lastly, the **Orders** table and the **Customers** table have a many-to-one (M:1) relationship (a customer can place multiple orders) with `CustomerID` acting as a foreign key, as shown in Figure 5.

3.2 Pivot Table Operations: Slicing, Dicing, and Drill-Down

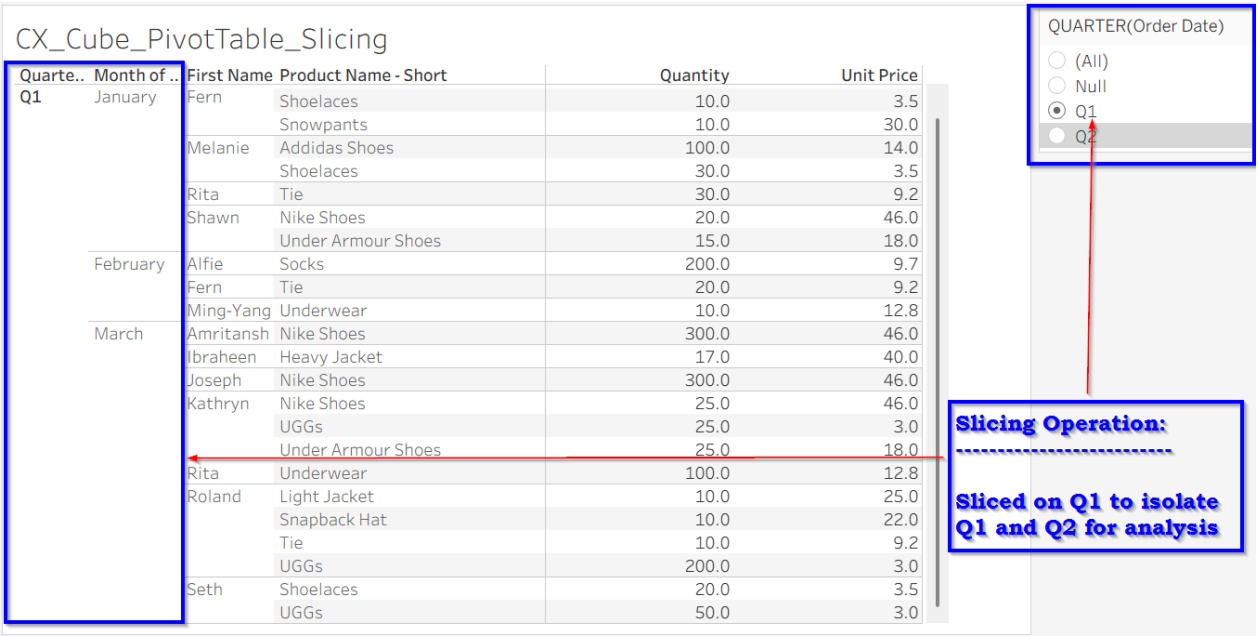
Pivot Tables were utilized to perform Online Analytical Processing (OLAP) on the Data Cube. The customer pivot table is illustrated in Figure 6.

- **Slicing with the Time Dimension:**
  For temporal analysis, the *Time* dimension (specifically `Order Date` organized by Quarter and Month) was positioned on the left-most axis of the pivot table Figure 7. This arrangement allowed purchasing behaviors to be "sliced" to isolate performance in Q1 versus Q2.
- **Dicing and Drill-Down:**
  The data was "diced" by *Product dimension* using *Product Category* against these time slices. Through this high-dimensional view, information can be obtained, such as popular footwear and winter apparel in Q1, as illustrated in Figure 8. Repeated slicing and dicing operations constitute a "drill-down" process, enabling the isolation of specific, actionable data. The **drill-down** operation illustrated in Figure 8 revealed the most popular products in footwear and winter apparel categories during **March**.



Quarte..	Month of ..	First Name	Product Name - Short	Quantity	Unit Price
Q1	March	Joseph	Nike Shoes	300.0	46.0
		Kathryn	Nike Shoes	25.0	46.0
			UGGs	25.0	3.0
			Under Armour Shoes	25.0	18.0
		Rita	Underwear	100.0	12.8
		Roland	Light Jacket	10.0	25.0
			Snapback Hat	10.0	22.0
			Tie	10.0	9.2
			UGGs	200.0	3.0
		Seth	Shoelaces	20.0	3.5
			UGGs	50.0	3.0
Q2	April	Alfie	Fitted Hat	50.0	10.0
			Heavy Jacket	3.0	40.0
			Jeans	0.0	38.0
			UGGs	0.0	3.0
		Amritansh	Dress Shirt	50.0	18.4
			Socks	50.0	9.7
		Ernie	Dress Shirt	30.0	18.4
			Snowboots	25.0	21.4
			Socks	30.0	9.7
		Fern	Blazer	40.0	81.0
			Jeans	10.0	38.0
			Short Sleeve Shirt	40.0	7.0

Figure 6: Pivot Table



Quarte..	Month of ..	First Name	Product Name - Short	Quantity	Unit Price
Q1	January	Fern	Shoelaces	10.0	3.5
			Snowpants	10.0	30.0
		Melanie	Addidas Shoes	100.0	14.0
			Shoelaces	30.0	3.5
		Rita	Tie	30.0	9.2
		Shawn	Nike Shoes	20.0	46.0
			Under Armour Shoes	15.0	18.0
	February	Alfie	Socks	200.0	9.7
		Fern	Tie	20.0	9.2
		Ming-Yang	Underwear	10.0	12.8
	March	Amritansh	Nike Shoes	300.0	46.0
		Ibraheen	Heavy Jacket	17.0	40.0
		Joseph	Nike Shoes	300.0	46.0
		Kathryn	Nike Shoes	25.0	46.0
			UGGs	25.0	3.0
			Under Armour Shoes	25.0	18.0
		Rita	Underwear	100.0	12.8
		Roland	Light Jacket	10.0	25.0
			Snapback Hat	10.0	22.0
			Tie	10.0	9.2
			UGGs	200.0	3.0
		Seth	Shoelaces	20.0	3.5
			UGGs	50.0	3.0

Figure 7: Customer: Pivot Table - Slicing Operation

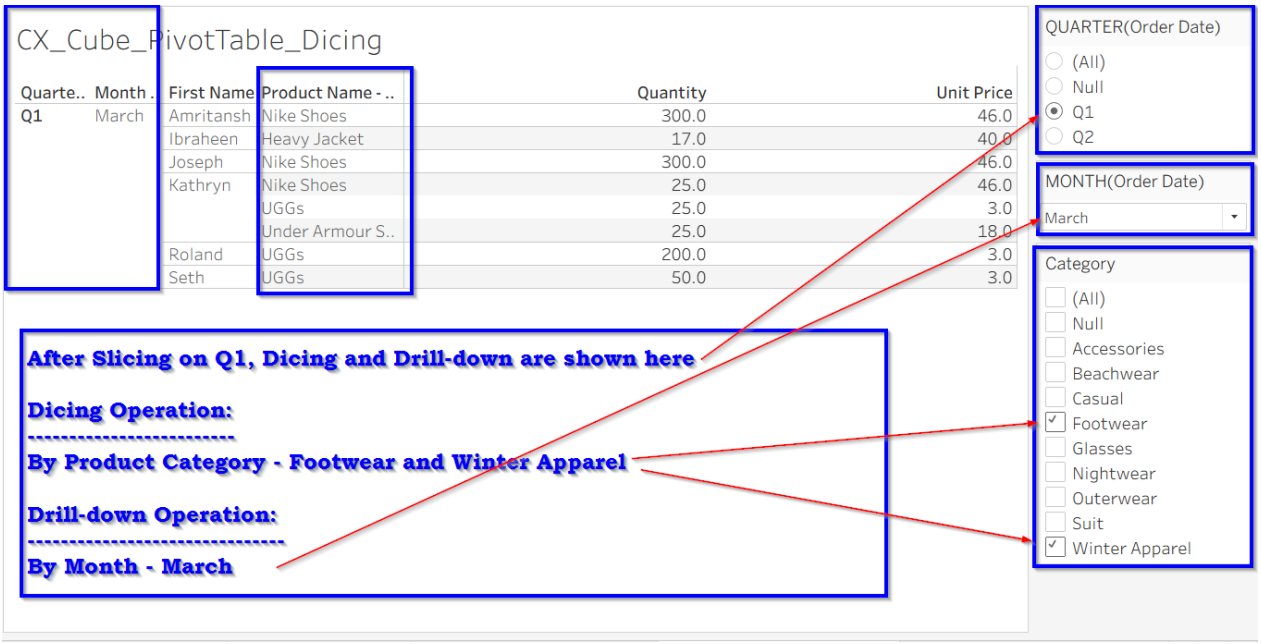


Figure 8: Customer: Pivot Table - Dicing Operation

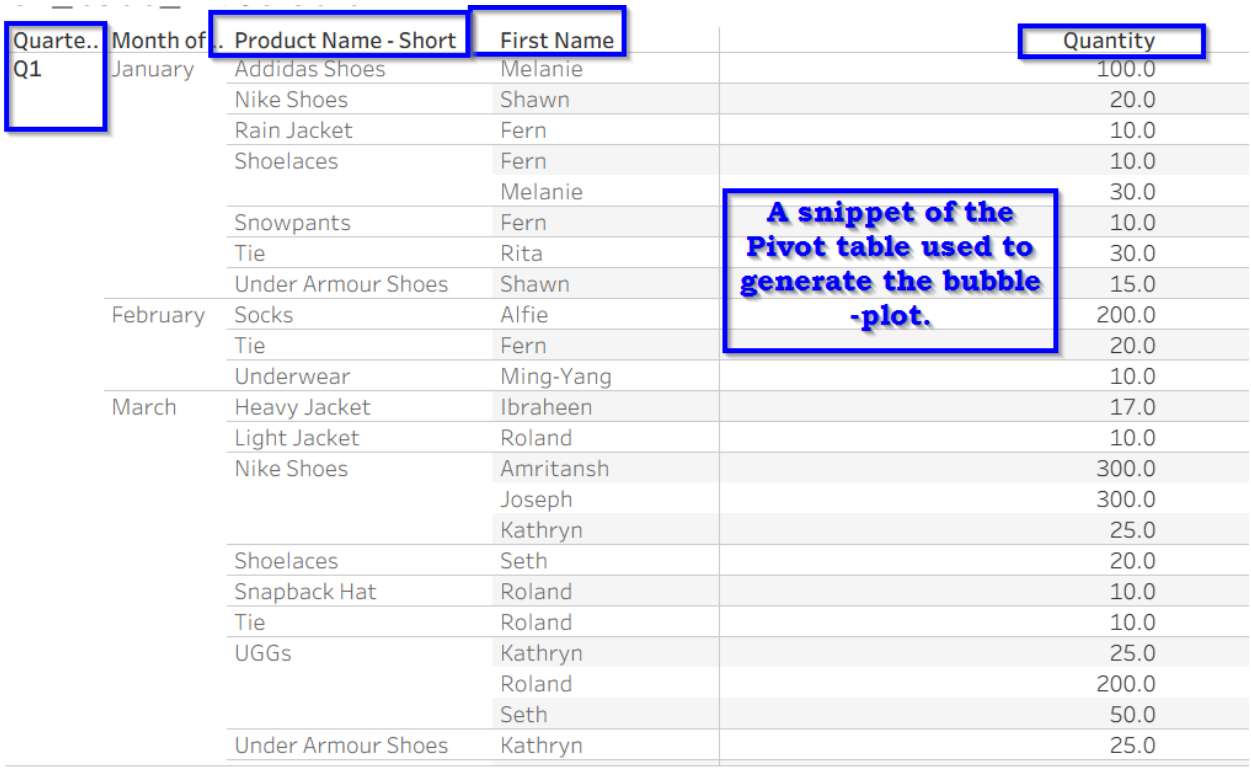


Figure 9: High-Dimensional Pivot Table - Fact Data Quantity

3.3 Visualization and Strategic Findings

To communicate these insights effectively, three distinct visualizations were generated. The underlying data tables for these charts are provided in the appendix (See section 7) to substantiate the findings.

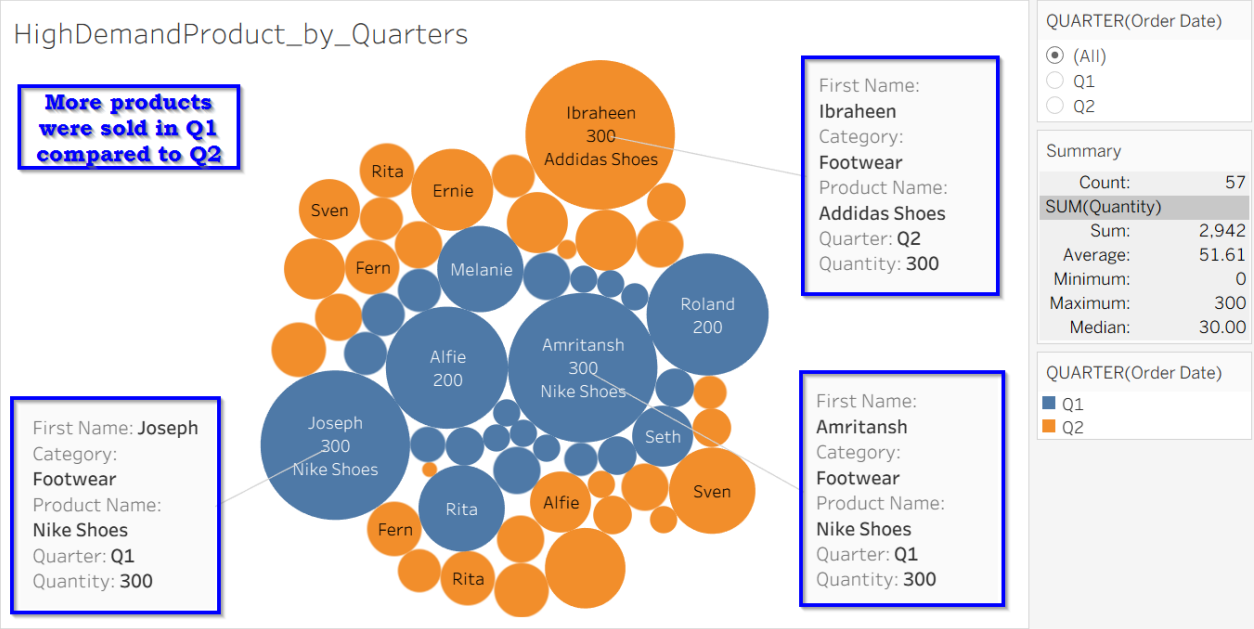


Figure 10: Bubble Plot - High demand product (quantity) & Customer who purchased the product

3.3.1 Bubble Chart

To understand the demand of a certain product and its popularity among customers over two quarters, a Bubble Plot was generated.

- **Dimensions Visualized:**
  Customer Name, Category Name, Product Name, Quarter, and Quantity (See Figure 10). The data related to this plot is provided in a screenshot of the table. (See Figure 9)
- **Finding:**
  It was observed that "Amritansh" and "Joseph" drove bulk demand in Q1, both purchasing **300 units of Nike Shoes** in **Q1**. In Q2, "Ibraheen" was the **highest-volume purchaser**, acquiring **300 units of Adidas Shoes**. This suggests a seasonal bulk-buying pattern rather than steady consumption.

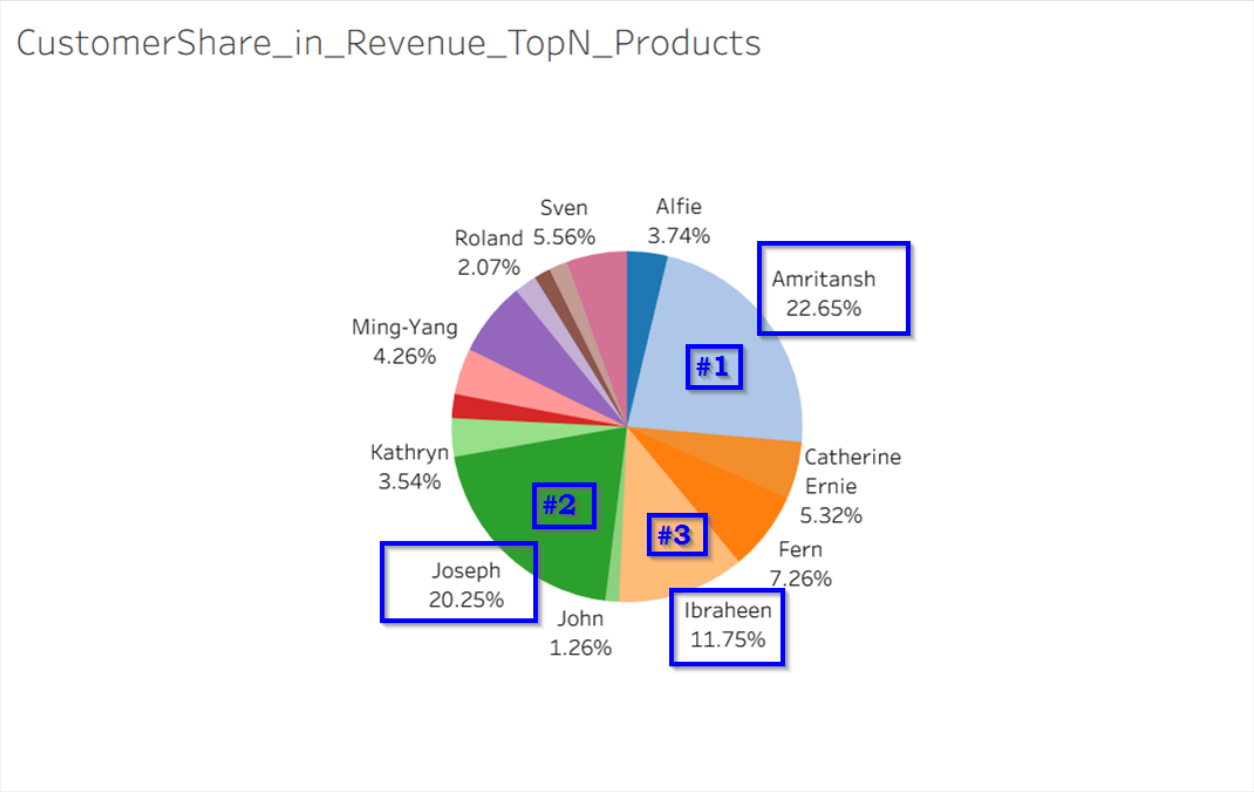


Figure 11: Customer: Revenue Contribution - Pie Chart

3.3.2 Revenue Dependence Risk (Pie Chart)

To address the strategic goal of identifying "Key Account Risk," the total revenue contribution per customer was analyzed and illustrated in Figure 11.

- **Visualization:**
  A Pie Chart illustrating the percentage share of Total Revenue was created (See Table 1 for the source dataset).
- **Finding:**
  A dangerous lack of diversification was revealed. It was found that three customers alone — **Amritansh (22.65%)**, **Joseph (20.25%)**, and **Ibraheen (11.75%)** — account for more than half of the total revenue. This confirms a high-risk scenario where the loss of a single customer among the top-3 could destabilize the company financially.

3.3.3 Advanced Pareto Analysis: Customer Portfolio Concentration

To mathematically validate the strategic risk of "Revenue Dependence," a dual-axis Pareto analysis was implemented (See Figure 12). This visualization correlates the cumulative percentage of the customer base against the cumulative percentage of total revenue.

Methodology:
Window calculations were utilized in Tableau to derive the Lorenz curve [4] coordinates.

- **Cumulative Revenue (Y-Axis):**
  The metric was calculated using the formula:

$$\text{ParetoY}(\%\text{Revenue}) = \frac{\text{RUNNING\_SUM}(\sum[\text{Revenue}])}{\text{TOTAL}(\sum[\text{Revenue}])} \tag{1}$$

- **Customer Percentile (X-Axis):**
  The metric was calculated using the index ratio over the sorted Customer dimension.

$$\text{ParetoX}(\%\text{Customer}) = \frac{\text{INDEX}()}{\text{SIZE}()} \tag{2}$$

Strategic Insight (The 80/25 Rule): As illustrated in the chart (See Figure 12), a significant concentration of financial value was observed within a minority of the client base.

- **The 50% Threshold:**
  It was identified that the **top 10.71%** of customers (represented by the marker for client "Ibraheen") generated **54.65%** of the total revenue.

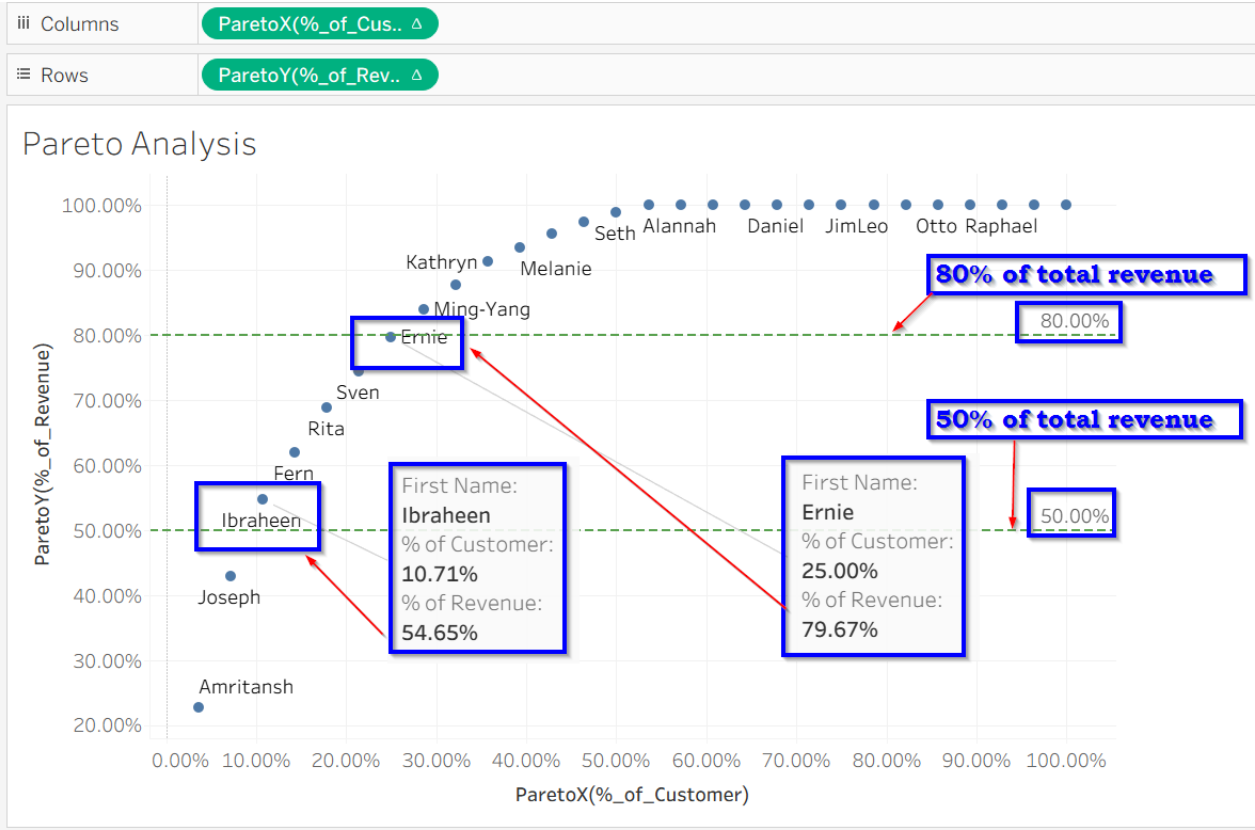


Figure 12: Customer: Pareto Analysis

- **The 80% Threshold:**
  The intersection lines confirmed that **top 25.00%** of the customer base (up to client "Ernie") accounts for **79.67%** of total revenue.

Strategic Connection:
This analysis validates the "Revenue Dependence Risk" identified in the BI Strategy. The finding that approximately 80% of revenue is derived from only a quarter of the customers indicates a high volatility risk; the loss of key accounts in this top quartile would have a disproportionate impact on financial stability.

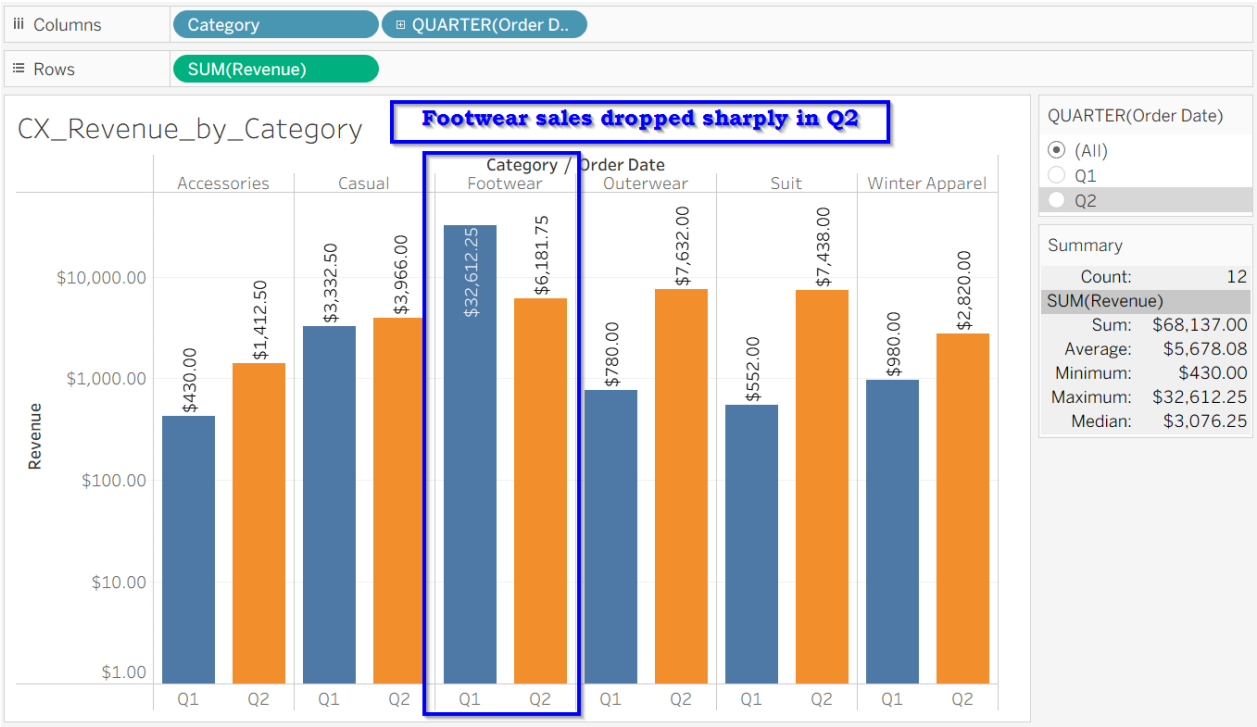


Figure 13: Revenue by Category for Q1 vs Q2

3.3.4 Product Portfolio Strength (Bar Chart)

Finally, the product mix was analyzed to determine which categories and which products were carrying the revenue load.

- **Visualization (Category):**
  A Bar Chart (See Figure 13) for Revenue generated by Category was produced (See Table 2 for the source dataset).
- **Findings (Category):**
  "Footwear" category was identified as the dominant revenue generator during Q1 significantly outperforming other categories. However, footwear sales dropped sharply

during Q2. All other categories generated sales during Q2.
The revenue generated during Q1 ($38,686.75) was still better than that of Q2 ($29,450.25), indicating that footwear was a significant part of the total revenue. An ample stock of footwear must be kept in the inventory during Q1 based on the demand.

- **Visualization (Product):**
  A Bar Chart (See Figure 14) for Revenue generated by Product was produced (See Table 3 for the source dataset).
- **Findings (Product):**
  "Nike Shoes" was identified as the dominant revenue generator ($29,670.00) during Q1, significantly outperforming other categories. "Adidas Shoes" was the best performing product ($5,418.00) in Q2. **Two products** (UGGs and Under Armour Shoes) recorded **zero sales volume in Q2**. Conversely, **11 products** were **introduced in Q2** that had no prior sales volume in Q1.

Observation:
There was a significant shift in product demand from Q1 to Q2, with the **Suits** & **Outerwear** categories showed increment in revenue in Q2, while **Footwear** sales decreased drastically.

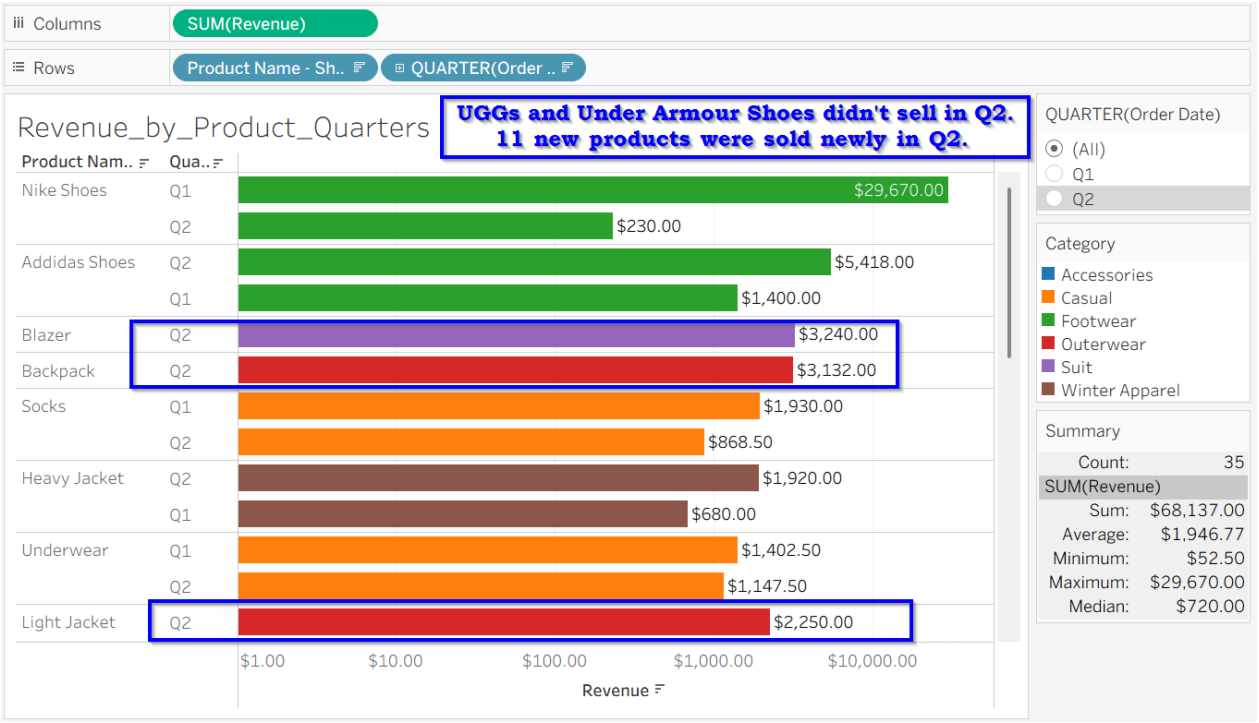


Figure 14: Revenue by Product for Q1 vs Q2

4 Supplier Side Analysis (Supply)

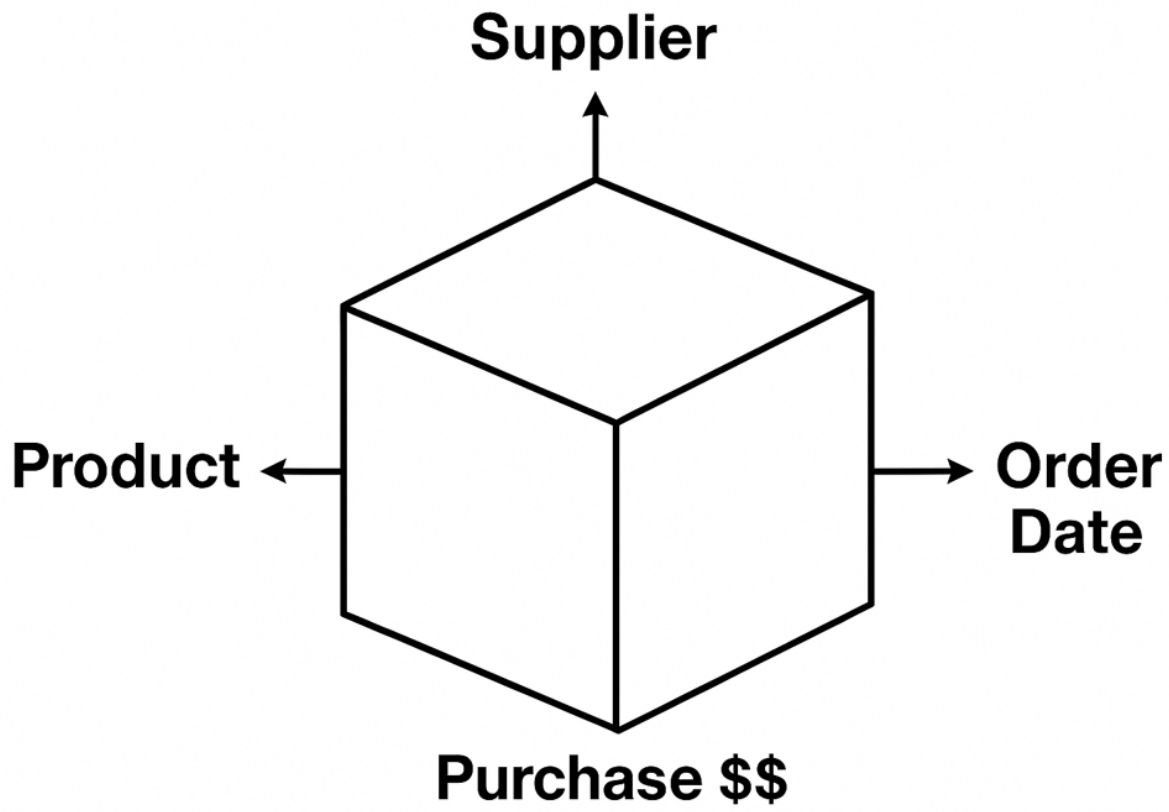


Figure 15: Supplier Data-cube: Supplier, Product, Submitted Date(Time)

In accordance with the Business Intelligence (BI) strategy, the analysis was extended to the supply side to identify cost-optimization opportunities and evaluate supplier pricing structures. This phase progressed from defining the data structure to high-dimensional exploration (using pivot tables), outlier detection using SQL, and ultimately to prescriptive modeling through statistical clustering.

4.1 Construction of the Supply-Side Data Cube

To facilitate this analysis, a specific Supply-Side Data Cube was constructed (See Figure 15). The Snowflake Schema was defined to capture procurement activities:

- **Fact Table:**
  `Purchase Order Details` (Facts: *Quantity* and *UnitCost*).
- **Dimension Tables:**
  `Suppliers` (Who), `Products` (What), `Purchase Orders` (When).

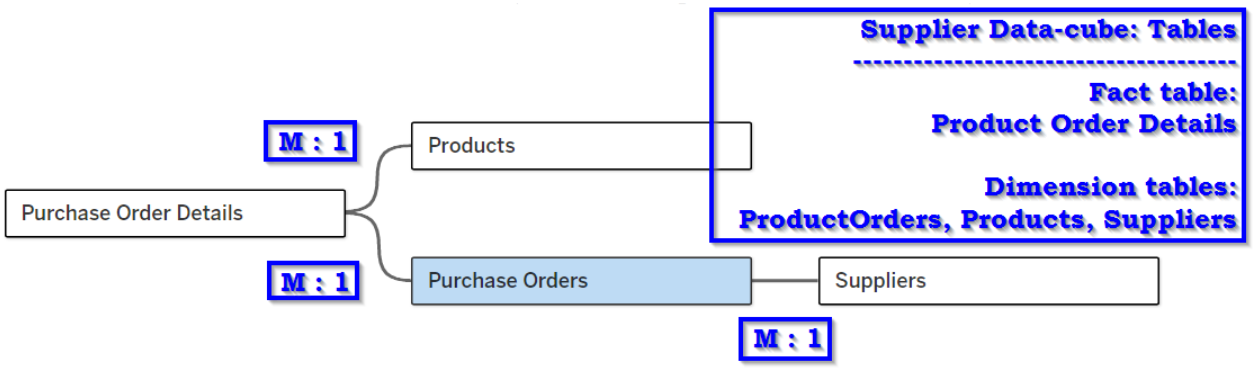


Figure 16: Supplier Data-cube: Snowflake Schema Tables

These tables were linked via Primary and Foreign Keys (e.g., `SupplierID`, `ProductID`) to create a multi-dimensional structure capable of supporting the slicing and dicing operations required to answer complex questions about supplies. The **Purchase Order Details** table and **Purchase Orders** table have a many-to-one (M:1) relationship, with `Purchase OrderID` acting as a foreign key for the **Purchase Order Details** table. The **Purchase Order Details** table and the **Products** table have a many-to-one (M:1) relationship, with `ProductID` acting as the foreign key for the **Purchase Order Details** table. Lastly, the **Purchase Orders** table and the **Suppliers** table have a many-to-one (M:1) relationship (a supplier can supply for multiple Product Orders) with `SupplierID` acting as a foreign key for the **Purchase Order Table**, as shown in Figure 16.

4.2 Pivot Table Operations: Slicing and Dicing

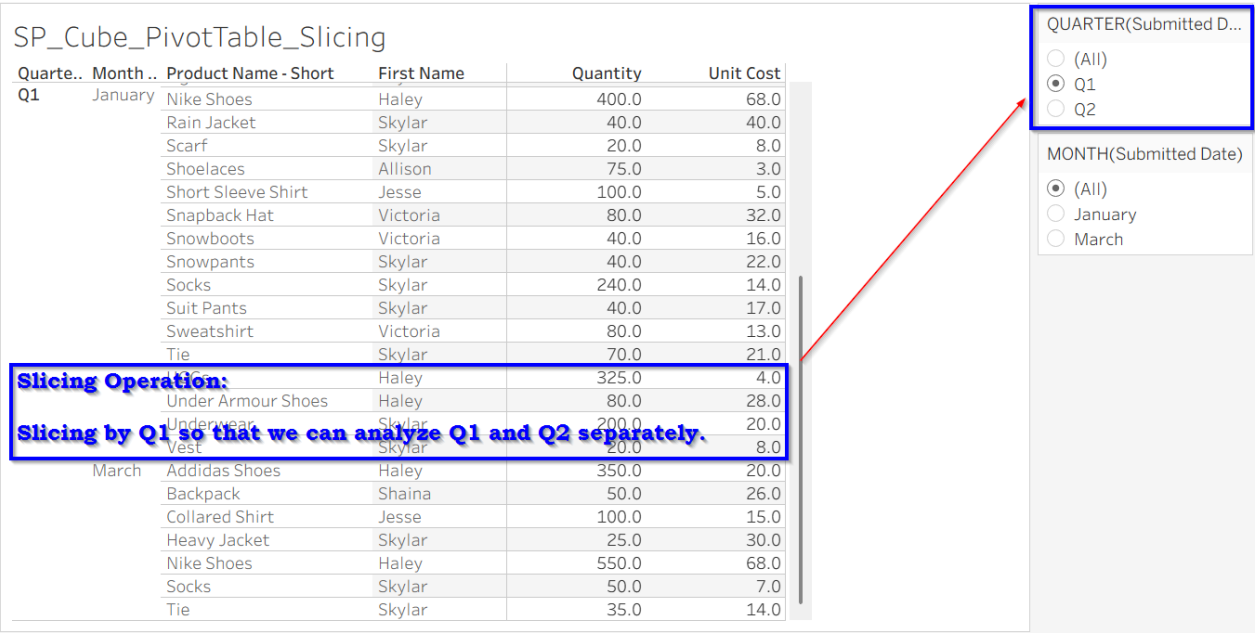
To investigate purchasing patterns, High-dimensional pivot tables were utilized to perform Online Analytical Processing (OLAP) operations.

Slicing and Dicing Strategy:
The data was sliced by the *Time dimension - Q1* (See Figure 17) and diced by *Product Category*. This operation enabled granular verification of individual procurement transactions. For example, through dicing and drill-down operations, it was determined that 350 units of "Adidas Shoes" were procured from "Haley" in Q1 at a unit price of $20.00. (See Figure 18).

4.3 Price Variance Identification (SQL Queries)

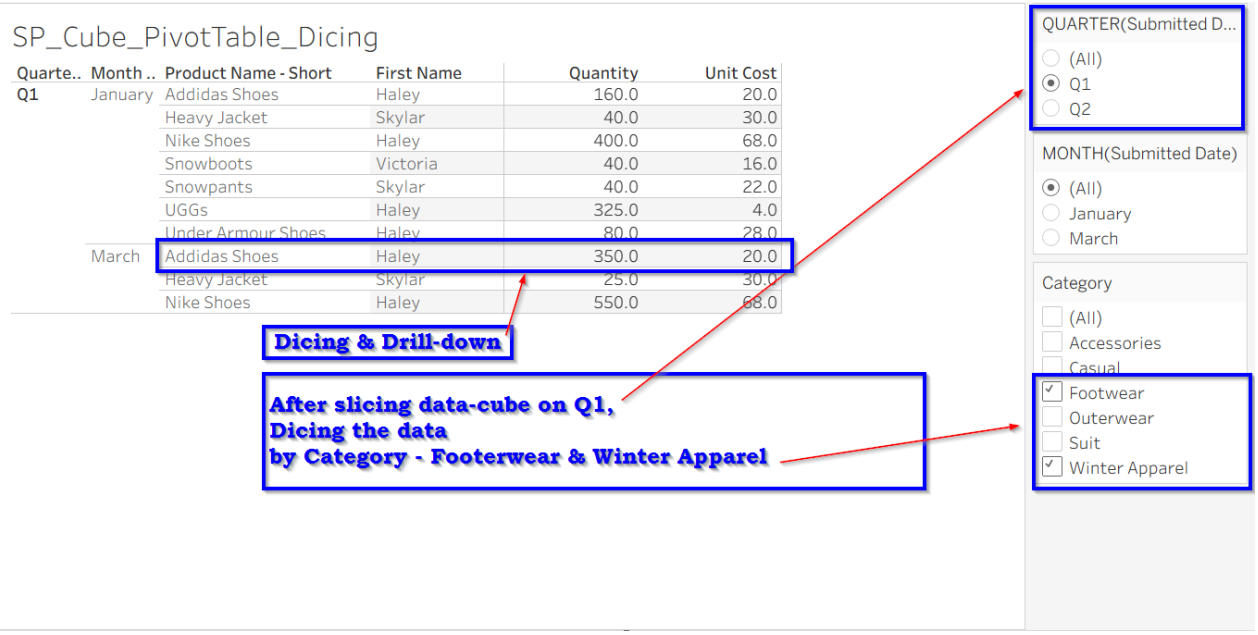
To address the strategic objective of identifying overcharging suppliers, a hybrid analytical approach integrating pivot tables, Python scripting, and SQL logic was implemented.

Methodology:
The price variance analysis was executed through the following multi-stage workflow:



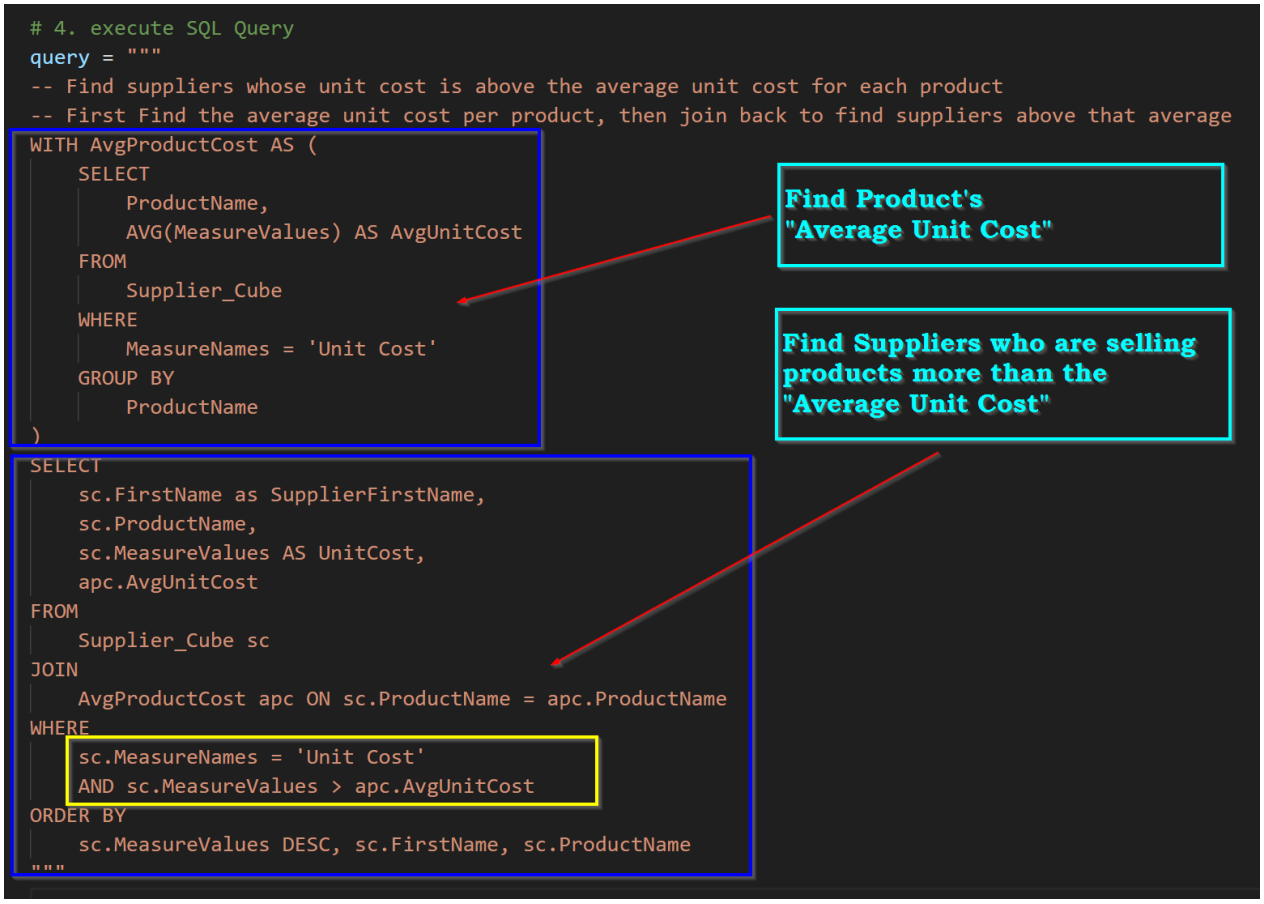
Quarte..	Month ..	Product Name - Short	First Name	Quantity	Unit Cost
Q1	January	Nike Shoes	Haley	400.0	68.0
		Rain Jacket	Skylar	40.0	40.0
		Scarf	Skylar	20.0	8.0
		Shoelaces	Allison	75.0	3.0
		Short Sleeve Shirt	Jesse	100.0	5.0
		Snapback Hat	Victoria	80.0	32.0
		Snowboots	Victoria	40.0	16.0
		Snowpants	Skylar	40.0	22.0
		Socks	Skylar	240.0	14.0
		Suit Pants	Skylar	40.0	17.0
		Sweatshirt	Victoria	80.0	13.0
		Tie	Skylar	70.0	21.0
		UGGs	Haley	325.0	4.0
		Under Armour Shoes	Haley	80.0	28.0
		Underwear	Skylar	200.0	20.0
		Vest	Skylar	20.0	8.0
	March	Addidas Shoes	Haley	350.0	20.0
		Backpack	Shaina	50.0	26.0
		Collared Shirt	Jesse	100.0	15.0
		Heavy Jacket	Skylar	25.0	30.0
		Nike Shoes	Haley	550.0	68.0
		Socks	Skylar	50.0	7.0
		Tie	Skylar	35.0	14.0

Figure 17: Supplier: Pivot Table - Slicing Operation



Quarte..	Month ..	Product Name - Short	First Name	Quantity	Unit Cost
Q1	January	Addidas Shoes	Haley	160.0	20.0
		Heavy Jacket	Skylar	40.0	30.0
		Nike Shoes	Haley	400.0	68.0
		Snowboots	Victoria	40.0	16.0
		Snowpants	Skylar	40.0	22.0
		UGGs	Haley	325.0	4.0
		Under Armour Shoes	Haley	80.0	28.0
	March	Addidas Shoes	Haley	350.0	20.0
		Heavy Jacket	Skylar	25.0	30.0
		Nike Shoes	Haley	550.0	68.0

Figure 18: Supplier: Pivot Table - Dicing Operation




```
# 4. execute SQL Query
query = """
-- Find suppliers whose unit cost is above the average unit cost for each product
-- First Find the average unit cost per product, then join back to find suppliers above that average
WITH AvgProductCost AS (
    SELECT
        ProductName,
        AVG(MeasureValues) AS AvgUnitCost
    FROM
        Supplier_Cube
    WHERE
        MeasureNames = 'Unit Cost'
    GROUP BY
        ProductName
)
SELECT
    sc.FirstName as SupplierFirstName,
    sc.ProductName,
    sc.MeasureValues AS UnitCost,
    apc.AvgUnitCost
FROM
    Supplier_Cube sc
JOIN
    AvgProductCost apc ON sc.ProductName = apc.ProductName
WHERE
    sc.MeasureNames = 'Unit Cost'
    AND sc.MeasureValues > apc.AvgUnitCost
ORDER BY
    sc.MeasureValues DESC, sc.FirstName, sc.ProductName
"""
```

Figure 19: Analytics Technique - Query

1. **Data Extraction:**
   The processed data from the Supplier-Side Pivot Table was exported into a structured Excel datasheet (`Supplier_Cube_PivotTable_TT.xlsx`) using the Tableau Table extension. This step ensured that the data structure resulting from the previous slicing and dicing operations was preserved for algorithmic analysis.

2. **In-Memory Database Construction:**
   Python was utilized to ingest this Excel dataset into an in-memory relational database using the `sqlite3` library. This architecture was selected to allow for the execution of complex Structured Query Language (SQL) scripts on the static dataset without requiring direct write access to the enterprise warehouse. The Python script implementing this logic (`query.py`) and the source dataset are available in the project repository:
   https://github.com/bhavik-knight/5560-high-dimensional-analysis

3. **Common Table Expression (CTE) Logic:**
   A query was constructed using a Common Table Expression (CTE) to derive the benchmark costs. The CTE first calculated the *Average Unit Price* for every product across all vendors. This derived metric was then re-joined with the main dataset to isolate specific transactions where suppliers were charging more than the average unit cost (See Figure 19).

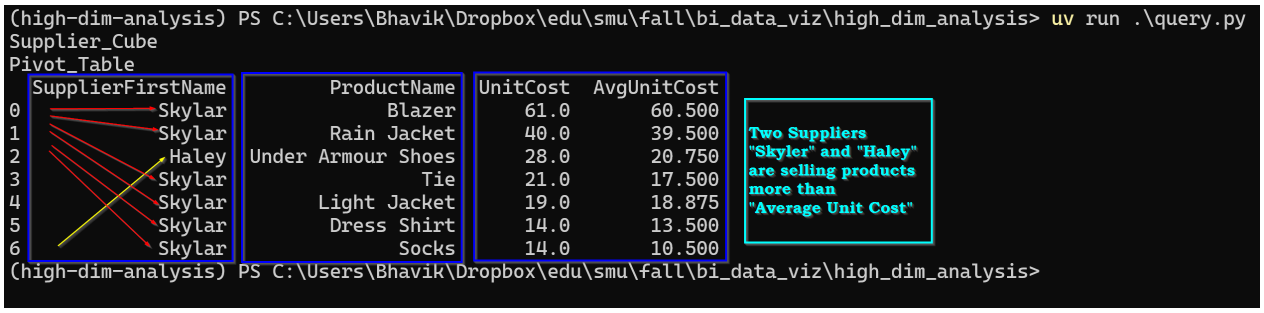


Figure 20: Analytics Technique - Query Result

Strategic Finding:
The query results (See Figure 20) indicated that specific suppliers, notably **Haley** and **Skylar**, consistently sold products such as “Under Armour Shoes” and “Light Jackets” at prices higher than the market average. This finding isolates specific targets for price negotiation to reduce procurement costs.

4.4 Supplier Substitution Analysis (Statistical Clustering)

To identify viable substitute suppliers for price negotiation, an unsupervised statistical classification method—**Hierarchical Agglomerative Clustering**—was applied to determine supplier similarity based on product overlap.

Methodology:
Since the dataset is categorical (representing the presence or absence of a product in a supplier's catalog), the data was first one-hot encoded (0/1). The hierarchical clustering algorithm is defined by **only two hyperparameters** (i.e., the distance metric and the linkage criterion). The data used for this statistical analysis are presented in the appendix Table 5.

- **Distance Metric:**
  The **Jaccard Distance** was selected rather than Euclidean distance. This metric was chosen because it focuses solely on the intersection of products provided (`1-1 matches`), ignoring the infinite combinations of products that neither supplier provides (`0-0 matches`).

  The formula for Jaccard distance is: $D(A, B) = 1 - J(A, B)$; where

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{3}$$

- **Linkage:**
  Average linkage criteria were applied to build the cluster hierarchy. [3]

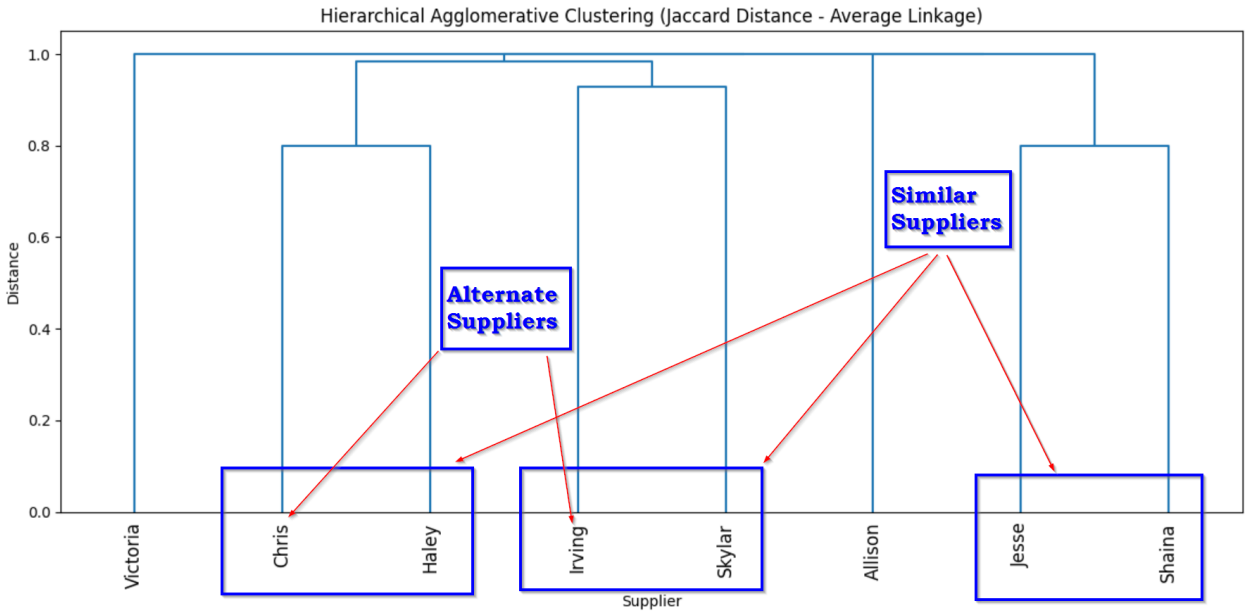


Figure 21: Analytics Technique - Hierarchical Clustering

Clustering Results and Recommendation: As illustrated in the resulting dendrogram (See Figure 21), distinct clusters of similarity were observed. Thus, negotiating with similar suppliers for lower prices could be beneficial for the business, as the same products can be procured at a cheaper rate.

1. **Chris over Haley:**
   It was observed that supplier "Haley" shares high similarity with supplier "Chris". Since the SQL analysis identified Haley as an expensive supplier, "Chris" is recommended as a cost-effective substitute for products like Under Armour Shoes ($13.50 / unit) compared to Haley ($28.00 / unit). The organization could realize savings of $1,160.00 **by shifting the procurement of these 80 units from Haley to Chris** (See Figure 22).

2. **Irving over Skylar:**
   Similarly, "Skylar" was found to be clustered with "Irving". Consequently, purchasing volume for Dress Shirts could be shifted from Skylar to Irving to achieve cost savings without sacrificing product availability. This could **save $1.00 per unit** should the supplier be switched to Irving instead of Skylar (See Figure 23).

3. **Chris over Skylar:**
   Although Chris and Skylar are not directly clustered together, **relaxing the clustering threshold** reveals similarity between 'Skylar' and 'Chris' at a higher hierarchical level. This could **save $25.00 per 100 units** if the supplier is to be switched to Chris instead of Skylar (See Figure 24).

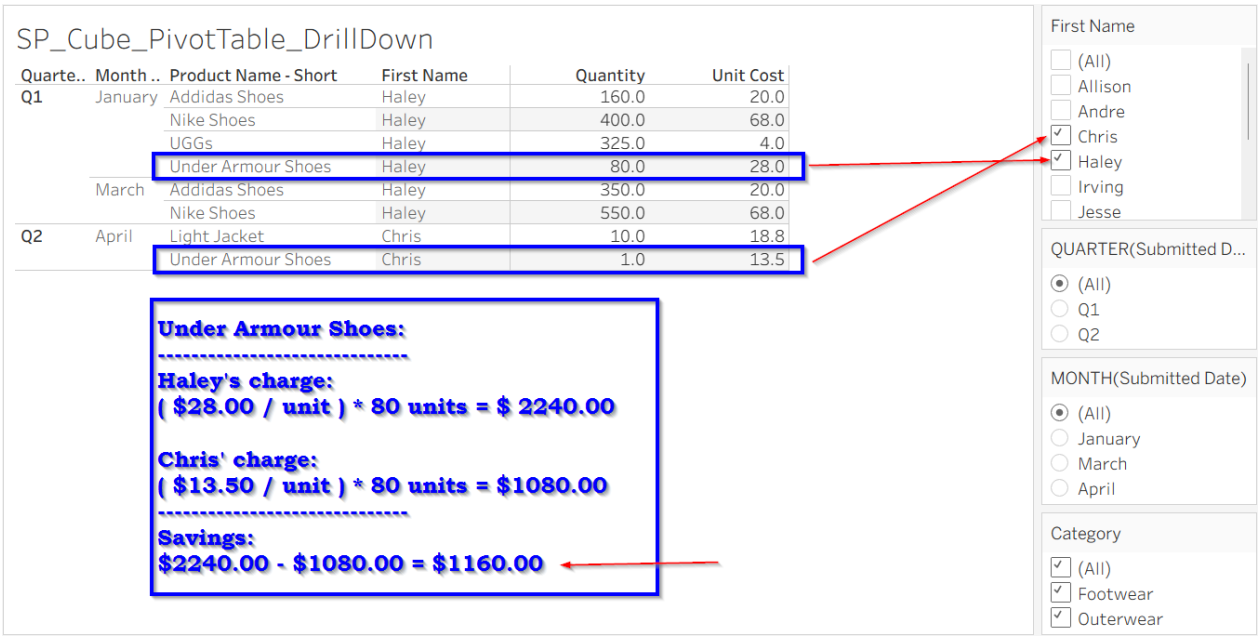
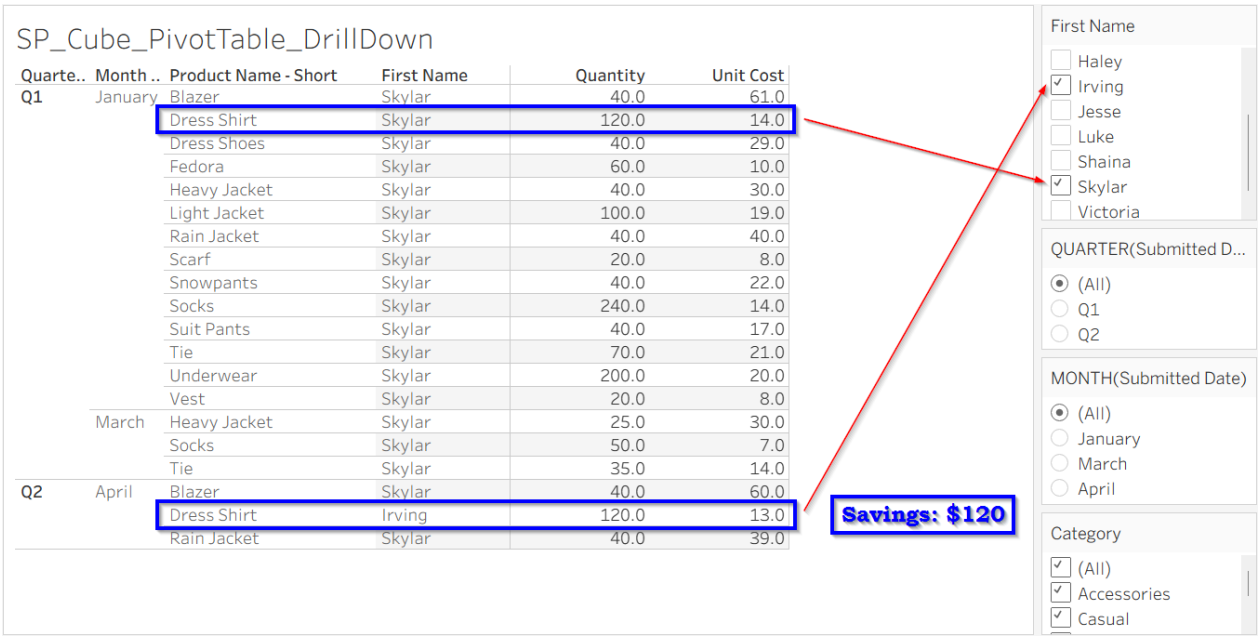


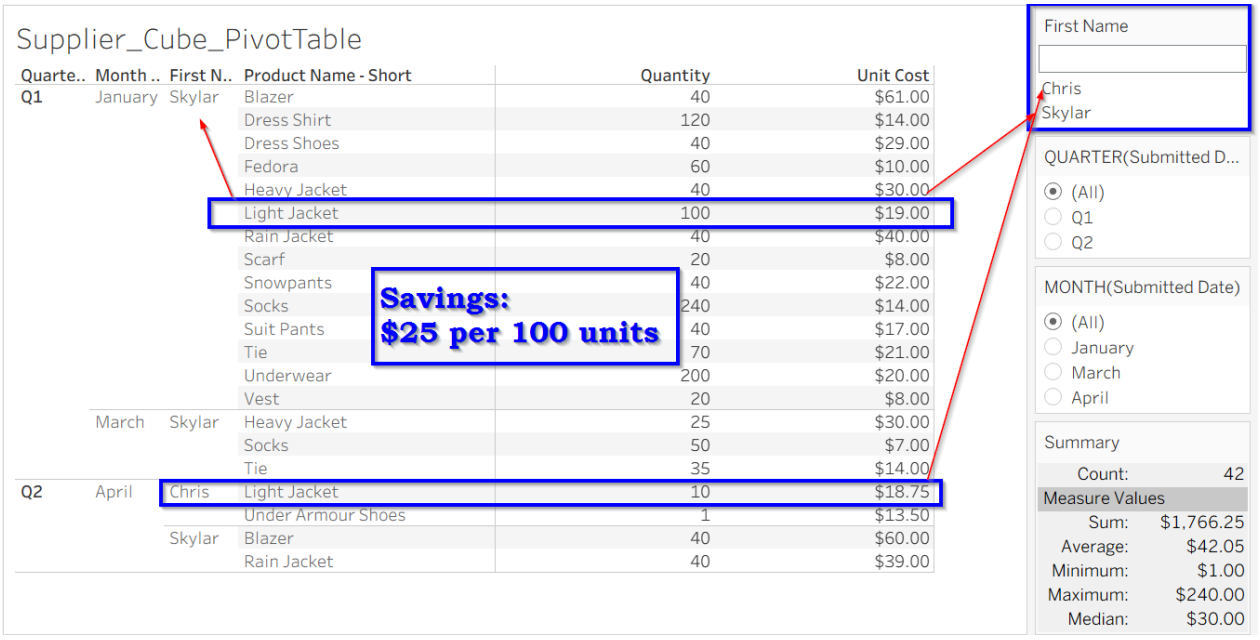
Figure 22: Analytics Technique - Hierarchical Clustering

A Python script is written to generate this hierarchical clustering. The data for which can be found in the appendix section Table 5.
Please visit the following link, and follow `README.md` file's instructions to run the script to reproduce the result.
https://github.com/bhavik-knight/5560-high-dimensional-analysis



Quarte..	Month ..	Product Name - Short	First Name	Quantity	Unit Cost
Q1	January	Blazer	Skylar	40.0	61.0
		Dress Shirt	Skylar	120.0	14.0
		Dress Shoes	Skylar	40.0	29.0
		Fedora	Skylar	60.0	10.0
		Heavy Jacket	Skylar	40.0	30.0
		Light Jacket	Skylar	100.0	19.0
		Rain Jacket	Skylar	40.0	40.0
		Scarf	Skylar	20.0	8.0
		Snowpants	Skylar	40.0	22.0
		Socks	Skylar	240.0	14.0
		Suit Pants	Skylar	40.0	17.0
		Tie	Skylar	70.0	21.0
		Underwear	Skylar	200.0	20.0
		Vest	Skylar	20.0	8.0
	March	Heavy Jacket	Skylar	25.0	30.0
		Socks	Skylar	50.0	7.0
		Tie	Skylar	35.0	14.0
Q2	April	Blazer	Skylar	40.0	60.0
		Dress Shirt	Irving	120.0	13.0
		Rain Jacket	Skylar	40.0	39.0

Figure 23: Analytics Technique - Hierarchical Clustering



Quarte..	Month ..	First N..	Product Name - Short	Quantity	Unit Cost
Q1	January	Skylar	Blazer	40	$61.00
			Dress Shirt	120	$14.00
			Dress Shoes	40	$29.00
			Fedora	60	$10.00
			Heavy Jacket	40	$30.00
			Light Jacket	100	$19.00
			Rain Jacket	40	$40.00
			Scarf	20	$8.00
			Snowpants	40	$22.00
			Socks	240	$14.00
			Suit Pants	40	$17.00
			Tie	70	$21.00
			Underwear	200	$20.00
			Vest	20	$8.00
	March	Skylar	Heavy Jacket	25	$30.00
			Socks	50	$7.00
			Tie	35	$14.00
Q2	April	Chris	Light Jacket	10	$18.75
			Under Armour Shoes	1	$13.50
		Skylar	Blazer	40	$60.00
			Rain Jacket	40	$39.00

Figure 24: Analytics Technique - Hierarchical Clustering

5 Story Summaries and Strategic Conclusions

This section synthesizes the analytical findings into actionable business insights. Each summary corresponds directly to the Business Intelligence (BI) strategies established in section 2, confirming that the high-dimensional analysis addressed the organization's critical business questions.

5.1 Customer Domain: Revenue Vulnerability and Seasonal Shifts

The customer-side analysis was driven by the strategic need to "evaluate revenue stability" and "track purchasing behaviors across time" (Strategies subsection 2.1).

- **Confirmation of Revenue Dependence Risk:**
  In alignment with the Pareto analysis strategy, the hypothesis regarding customer concentration was validated. It was determined that the organization faces a critical stability risk, as **54.65%** of total revenue is derived from **only three customers** *(Joseph, Amritansh, and Ibraheen)*. Furthermore, the dual-axis Pareto analysis confirmed that the **top 25% of customers** account for **nearly 80% of revenue**. The loss of any single key account would have a disproportionately severe financial impact, necessitating the development of a revenue diversification strategy.
- **Identification of Seasonal Inventory Cycles:**
  Through the high-dimensional slicing of the Product portfolio against the Time dimension (Strategy subsection 2.1), a significant temporal shift in demand was observed. While "Footwear" was identified as the dominant category in Q1 (generating $32,612), sales for this category dropped precipitously in Q2 (to $6,181). Conversely, categories such as "Suits" and "Outerwear" saw strong growth in Q2. This insight necessitates dynamic inventory planning, specifically reducing inventory levels of Nike Shoes while increasing stock of formal wear as the fiscal year progresses.

5.2 Supplier Domain: Cost Inefficiencies and Substitution Opportunities

The supplier-side analysis addressed the strategic objectives of "identifying price variances" and "determining substitute suppliers" via statistical clustering (Strategy subsection 2.2).

- **Detection of Overcharging Suppliers:**
  Pursuant to the SQL query strategy designed to flag costs above the market average, two specific suppliers—**Haley** and **Skylar**—were isolated as these vendors consistently charged prices higher than the average for identical items.
- **Prescriptive Supplier Substitution Analysis:**
  To resolve the pricing issue, the Hierarchical Clustering strategy was applied to identify similar vendors. Actionable supplier arbitrage opportunities were identified:

1. **Under Armour Shoes:**
   It is recommended that orders be shifted from Haley ($28.00/unit) to **Chris** ($13.50/unit), resulting in a savings of approximately $14.50 per unit.

2. **Dress Shirts:**
   It is recommended that orders be shifted from Skylar ($14.00/unit) to **Irving** ($13.00/unit) to achieve immediate marginal cost reductions without sacrificing product availability.

6 Learning Reflection

6.1 Moving from "Flat" to Multidimensional Data Analysis

A core theoretical takeaway from this assignment is the shift from flat-file data processing to high-dimensional rationality. As noted in the course theory, data analytics is not a black-box process but rather a "fishing pole" used to **extract targeted insights from the data lake**. [8]. Working with flat spreadsheets often obscures complex relationships due to data redundancy and "data-creator bias," where the analysis is limited to pre-selected columns.

By constructing **Star and/or Snowflake schemas** [7], we effectively enabled functional Data Cubes. This structural foundation was the prerequisite for resolving Many-to-Many relationships—such as Suppliers to Products—into usable One-to-Many relationships. This transition allowed us to move beyond simple aggregation to dynamic slicing, dicing, and drill-down [6], revealing that while "Footwear" drives revenue in Q1 ($32,612), demand collapses in Q2, necessitating a strategic inventory pivot to "Suits".

6.2 The Value of Defined Questions

Before analyzing the business problem, we learned the importance of defining clear questions and forming assumptions, rather than performing unguided exploratory analysis. Our *Business Intelligence Strategy* required us to formulate specific hypotheses, such as the existence of "Key Account Risk" and "Supplier Price Variance".

This question-driven approach validated that high-dimensional analysis supports crucial decision-making. By viewing data through the intersection of the *Customer Dimension* and *Revenue Measure*, we validated the Pareto principle [5]: a dangerous 54.65% of our revenue is dependent on just three customers (Amritansh, Joseph, and Ibraheen). This conclusion was not speculative, but rather a rational calculation.

6.3 The Synergy of Hybrid Analytical Techniques

Finally, this project demonstrated that a robust BI strategy relies on the integration of distinct analytical tools rather than a single method. We utilized a hybrid workflow to improve business processes:

6.3.1 Visualization (Descriptive Analysis)

Visualization tools provided the necessary descriptive foundation. **Bubble charts** [1], **Bar charts**, and **Pareto analysis** highlighted *where* the risks lay, specifically identifying the seasonal bulk-buying behaviors of top customers.

6.3.2 SQL Queries (Diagnostic Analysis)

SQL queries provided the analytical precision needed to isolate inefficiencies. We identified that suppliers *Haley* and *Skylar* were consistently overcharging for items like "Under Armour Shoes" compared to market averages.

6.3.3 Statistical Analysis (Prescriptive Analysis)

Moving from diagnosis to solution required statistical clustering. By applying **Hierarchical Agglomerative Clustering** [2] with **Jaccard distance**, we mathematically identified "Chris" and "Irving" as the most similar viable substitutes.

7 Appendix

7.1 Appendix A: Data and Supplemental Tables

7.1.1 Table 1: Customers' contribution in Total Revenue

Table 1: Detailed Customer Share of Total Revenue (All Listed Accounts)

First Name	Revenue ($)	% of Total Revenue
Amritansh	15,433.00	22.65%
Joseph	13,800.00	20.25%
Ibraheen	8,008.00	11.75%
Fern	4,949.00	7.26%
Rita	4,683.00	6.87%
Sven	3,787.00	5.56%
Ernie	3,625.00	5.32%
Ming-Yang	2,906.00	4.26%
Alfie	2,550.00	3.74%
Kathryn	2,411.00	3.54%
Melanie	1,505.00	2.21%
Roland	1,413.00	2.07%
Shawn	1,190.00	1.75%
Seth	1,020.00	1.50%
John	860.00	1.26%

7.1.2 Table 2: Total Revenue for each Product Category

Table 2: Total Revenue by each Category

Category	Q1 ($)	Q2 ($)
Accessories	430.00	1,412.50
Casual	3,332.50	3,966.00
Footwear	32,612.25	6,181.75
Outerwear	780.00	7,632.00
Suit	552.00	7,438.00
Winter Apparel	980.00	2,820.00

7.1.3 Table 3: Total Revenue for each Product

Table 3: Product Portfolio Performance by Category and Quarter (Revenue in $)

Category	Product	Q1 ($)	Q2 ($)
Accessories	Fitted Hat		500.00
	Scarf		200.00
	Shoelaces	210.00	52.50
	Snapback Hat	220.00	660.00
Casual	Collared Shirt		1,950.00
	Socks	1,930.00	868.50
	Underwear	1,402.50	1147.50
Footwear	Adidas Shoes	1,400.00	5,418.00
	Nike Shoes	29,679.00	230.00
	Snowboots		533.75
	UGGs	822.25	
	Under Armour Shoes	720.00	
Outerwear	Backpack		3,132.00
	Jeans		380.00
	Light Jacket	250.00	2,250.00
	Rain Jacket	530.00	1,590.00
	Short Sleeve Shirt		280.00
Suit	Blazer		3,240.00
	Dress Shirt		2,280.00
	Dress Shoes		1,560.00
	Tie	552.00	230.00
	Vest		200.00
Winter Apparel	Heavy Jacket	680.00	1,920.00
	Snowpants	300.00	900.00

7.1.4 Table 4: High-Dimensional Pivot Table - Supplier

Table 4: Detailed Sales Data: Quantity and Unit Cost by Quarter, Month, and Supplier (Multi-page format)

Quarter	Month	Supplier Name	Product	Quantity	Unit Cost
Q1	January	**Allison**	Shoelaces	75	3.00
		Haley	Adidas Shoes	160	20.00
			Nike Shoes	400	68.00
			UGGs	325	4.00
			Under Armour Shoes	80	28.00
		Jesse	Collared Shirt	80	15.00
			Jeans	120	28.00
			Short Sleeve Shirt	100	5.00
		Shaina	Backpack	40	26.00
		Skylar	Blazer	40	61.00
			Dress Shirt	120	14.00
			Dress Shoes	40	29.00
			Fedora	60	10.00
			Heavy Jacket	40	30.00
			Light Jacket	100	19.00
			Rain Jacket	40	40.00
			Scarf	20	8.00
			Snowpants	40	22.00
			Socks	240	14.00
			Suit Pants	40	17.00
			Tie	70	21.00
			Underwear	200	20.00
			Vest	20	8.00
		Victoria	Fitted Hat	150	16.00
			Football Jersey	40	16.00
			Snapback Hat	80	32.00
			Snowboots	40	16.00
			Sweatshirt	80	13.00
	March	**Haley**	Adidas Shoes	350	20.00
			Nike Shoes	550	68.00
		Jesse	Collared Shirt	100	15.00
		Shaina	Backpack	50	26.00
		Skylar	Heavy Jacket	25	30.00
			Socks	50	7.00
			Tie	35	14.00

Continued on next page

Table 4 – continued from previous page

Quarter	Month	Supplier Name	Product	Quantity	Unit Cost
Q2	April	**Chris**	Light Jacket	10	18.75
			Under Armour Shoes	1	13.50
		Irving	Dress Shirt	120	13.00
		Jesse	Backpack	40	26.00
		Shaina	Silver Necklace	10	9.00
		Skylar	Blazer	40	60.00
			Rain Jacket	40	39.00

7.1.5 Table 5: Suppliers with their supplied product Listings

Table 5: Supplier-Product Mapping

Supplier Name	Product ID
Haley	1
Haley	34
Haley	43
Haley	81
Skylar	6
Skylar	7
Skylar	8
Skylar	14
Skylar	17
Skylar	19
Skylar	20
Skylar	21
Skylar	40
Skylar	41
Skylar	48
Skylar	51
Skylar	74
Skylar	77
Victoria	3
Victoria	4
Victoria	5
Victoria	65
Victoria	66

Continued on next page

Table 5 – continued from previous page

Supplier Name	Product ID
Allison	80
Jesse	52
Jesse	56
Jesse	57
Jesse	72
Shaina	72
Shaina	85
Irving	40
Chris	1
Chris	6

7.2 Appendix B: Python Scripts

7.2.1 Script for SQL Query

Listing 1: SQL Query to Identify Overcharging Suppliers

```
import pandas as pd
import sqlite3

# 1. Read all sheets from Excel into a dictionary of DataFrames
excel_file = "Supplier_Cube_PivotTable_TT.xlsx"
sheets_dict = pd.read_excel(excel_file, sheet_name=None)

# 2. Create an in-memory SQLite database
conn = sqlite3.connect(":memory:")

# 3. Write each sheet to SQLite as a table
for sheet_name, df in sheets_dict.items():
    df.to_sql(sheet_name, conn, if_exists="replace", index=False)

# 4. SQL query using Common Table Expression (CTE)
query = """
WITH␣AvgProductCost␣AS␣(
␣␣␣␣SELECT
␣␣␣␣␣␣␣␣ProductName,
␣␣␣␣␣␣␣␣AVG(MeasureValues)␣AS␣AvgUnitCost
␣␣␣␣FROM
␣␣␣␣␣␣␣␣Supplier_Cube
␣␣␣␣WHERE
␣␣␣␣␣␣␣␣MeasureNames␣=␣'Unit␣Cost'
␣␣␣␣GROUP␣BY
␣␣␣␣␣␣␣␣ProductName
)
SELECT
␣␣␣␣sc.FirstName␣AS␣SupplierFirstName,
␣␣␣␣sc.ProductName,
␣␣␣␣sc.MeasureValues␣AS␣UnitCost,
␣␣␣␣apc.AvgUnitCost
FROM
␣␣␣␣Supplier_Cube␣sc
JOIN
␣␣␣␣AvgProductCost␣apc
ON
␣␣␣␣sc.ProductName␣=␣apc.ProductName
WHERE
␣␣␣␣sc.MeasureNames␣=␣'Unit␣Cost'
```

```
    AND sc.MeasureValues > apc.AvgUnitCost
ORDER BY
    sc.MeasureValues DESC,
    sc.FirstName,
    sc.ProductName;
"""

# 5. Execute query and display results
result_df = pd.read_sql_query(query, conn)
print(result_df)
```

7.2.2 Script for Statistical Analysis - Clustering

Listing 2: Hierarchical Clustering of Suppliers Using Jaccard Distance

```
import pandas as pd
import matplotlib.pyplot as plt
from scipy.spatial.distance import pdist
from scipy.cluster.hierarchy import linkage, dendrogram

# 1. Load dataset
file_path = "dataset2.xlsx"
df = pd.read_excel(file_path)

# 2. One-hot encode ProductID
df_encoded = pd.get_dummies(df, columns=["ProductID"])

# 3. Build Supplier x Product matrix
supplier_product_matrix = df_encoded.groupby("SupplierID").max()

# 4. Convert to NumPy matrix
X = supplier_product_matrix.values

# 5. Compute Jaccard distance
jaccard_dist = pdist(X, metric="jaccard")

# 6. Perform hierarchical clustering (average linkage)
Z = linkage(jaccard_dist, method="average")

# 7. Plot dendrogram
title = "Hierarchical Agglomerative Clustering"
title += " (Jaccard Distance, Average Linkage)"

plt.figure(figsize=(12, 6))
dendrogram(
```

```
    Z,
    labels=supplier_product_matrix.index.astype(str),
    leaf_rotation=90
)

plt.title(title)
plt.xlabel("Supplier")
plt.ylabel("Distance")
plt.tight_layout()
plt.savefig("dendrogram.png", bbox_inches="tight")
plt.show()
```

Bibliography

References

[1] Wikipedia contributors. **Bubble chart**. Wikipedia, The Free Encyclopedia. URL: https://en.wikipedia.org/wiki/Bubble_chart (visited on 12/13/2025) (cit. on p. 28).

[2] Wikipedia contributors. **Hierarchical clustering**. Wikipedia, The Free Encyclopedia. URL: https://en.wikipedia.org/wiki/Hierarchical_clustering (visited on 12/13/2025) (cit. on p. 29).

[3] Wikipedia contributors. **Jaccard index**. Wikipedia, The Free Encyclopedia. URL: https://en.wikipedia.org/w/index.php?title=Jaccard_index&oldid=1311865137 (visited on 12/13/2025) (cit. on p. 23).

[4] Wikipedia contributors. **Lorenz curve**. Wikipedia, The Free Encyclopedia. URL: https://en.wikipedia.org/w/index.php?title=Lorenz_curve&oldid=1323949549 (visited on 12/12/2025) (cit. on p. 14).

[5] Wikipedia contributors. **Pareto principle**. Wikipedia, The Free Encyclopedia. URL: https://en.wikipedia.org/wiki/Pareto_principle (visited on 12/13/2025) (cit. on p. 28).

[6] GeeksforGeeks. **Data Cube or OLAP Approach in Data Mining**. GeeksforGeeks. URL: https://www.geeksforgeeks.org/data-analysis/data-cube-or-olap-approach-in-data-mining/ (visited on 12/13/2025) (cit. on p. 28).

[7] GeeksforGeeks. **Difference between Star Schema and Snowflake Schema**. GeeksforGeeks. URL: https://www.geeksforgeeks.org/dbms/difference-between-star-schema-and-snowflake-schema/ (visited on 12/13/2025) (cit. on p. 28).

[8] GeeksforGeeks. **Multidimensional Data Model**. GeeksforGeeks. URL: https://www.geeksforgeeks.org/software-engineering/multidimensional-data-model/ (visited on 12/13/2025) (cit. on p. 28).

[9] Neri Van Otten. **How To Handle High-Dimensional Data In Machine Learning**. Spot Intelligence. 2024. URL: https://spotintelligence.com/2024/11/14/handling-high-dimensional-data/ (visited on 12/13/2025) (cit. on p. 3).